

Appln. No. : 10/054,051
Applicant : Epstein, *et al.*
Filed : January 22, 2002
TC/A.U. : 2121
Examiner : Hiri, Joseph P.

Confirmation No. 1351

Docket No. : BOC9-2000-0039 (180)

DECLARATION UNDER 37 C.F.R. § 1.131

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22203-1450

Sir:

I, Jerome L. Quinn, a citizen of the United States of America, residing at 2 Westview Avenue, North Salem, NY 10560, hereby declare and state as follows:

6. I was employed by International Business Machines Corporation (IBM) of Yorktown Heights, New York at the time the above-identified application was conceived and I continue to be employed by IBM. I make this declaration in support the above-identified application.

7. IBM has invested substantial time and effort into the research, development, and marketing of their products, and in an effort to protect its rights in all new inventions, IBM requests that all employees prepare and submit IBM Confidential Invention Disclosure Forms upon conception by the inventor(s).

8. As a named co-inventor for this invention, I submitted the attached IBM Confidential Invention Disclosure BOC8-2000-0029.

9. I make this Declaration to establish that I and my co-inventor Mark E. Epstein, conceived of the present invention at least as early as April 13, 2000, and exercised due diligence from prior to April 13, 2000 to January 22, 2002, the filing date for the above-identified patent application.

10. I further declare that all statements made herein of my own knowledge are true and all statements made on information and belief are believed to be true and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment or both under Section 1001 of Title 18 of the United States Code, and that such willful, false statements may jeopardize the validity of the above-identified patent application or any patent issuing thereon.

Jerome L. Quinn
Jerome L. Quinn

Date: June 14, 2004

STATE OF NEW YORK)
) ss:
COUNTY OF WESTCHESTER)


The foregoing instrument was sworn to and subscribed before me this 18th day of June, 2004, by JEROME L. QUINN, who is personally known to me or who has produced Driver's license (type of identification) as identification.

Robin Louise Moro
NOTARY PUBLIC,
STATE OF NEW YORK

(Print, Type or Stamp Commissioned Name of Notary Public)

Robin Louise Moro
Notary Public, State of NY
No. 01D16045019
County of Westchester
Commission Expires July 17, 2008

CONFIDENTIAL

	Disclosure BOC8-2000-0029
	Created By: Mark E Epstein Created On: 04/01/2000 09:34:44 AM
	Last Modified By: wpts1 wpts1 Last Modified On: 04/13/2000 09:13:29 PM
	*** IBM Confidential ***

Required fields are marked with the asterisk (*) and must be filled in to complete the form .

Summary

Status	Under Evaluation
Processing Location	BOC
Functional Area	Speech Development & Customization (O. Osborne)
Attorney/Patent Professional	Richard Tomlin/Boca Raton/IBM
IDT Team	[REDACTED]
Submitted Date	04/01/2000 11:14:49 AM
Owning Division	SWG
Serial	41
Incentive Program	
Lab	
Technology Code	

Inventors with Lotus Notes IDs

Inventors: Mark E Epstein/Watson/IBM

Inventor Name > denotes primary contact	Inventor Serial	Div/Dept	Manager Serial	Manager Name
> Epstein, Mark E.	[REDACTED]	[REDACTED]	[REDACTED]	[REDACTED]

Inventors without Lotus Notes IDs

IDT Selection

IDT Team: [REDACTED]	Attorney/Patent Professional: [REDACTED]
-------------------------	---

Response Due to IP&L : 05/13/2000

Main Idea

***Title of disclosure (In English)**

A Fast Way of Tuning Decision Tree Models

***Idea of disclosure**

1. Describe your invention, stating the problem solved (if appropriate), and indicating the advantages of using the invention.

Invention: The invention has 4 parts:

- Determine the potential usefulness of a new decision tree question without requiring the complete tree to be regrown.
- Determine where in a decision tree the correct answer would be found, and show where the tree went awry for a particular error.
- The GUI elements for viewing and debugging a decision tree.

JUL 26 2000

- The ability to regrow only parts of a decision tree.

Problem solved: Tuning decision tree models takes significant amount of time for an experienced developer. In the current VVT NLU 1.2 toolkit, the application developer tunes a decision tree by iteratively applying the following process:

- 1) Run a regression test on some training data with the current model.
- 2) Find a sentence that has an incorrect parse tree as output.
- 3) Locate where in the sentence the incorrect parse deviated from the truth.
- 4) Examine the training data to discover a feature that is different in the parse trees similar to the correct parse and similar to the incorrect parse.
- 5) Add a decision tree question which utilizes this feature.
- 6) Retrain the decision tree models.
- 7) Rerun the regression test.

For the current VVT NLU toolkit, it can take 3-7 minutes to collect the decision tree data, grow the trees, and smooth the trees. It then can take another 1-2 minutes to run the regression test. Thus, after the user spends possibly 1-15 minutes discovering the feature (steps 2-4), he usually has to spend 5-10 minutes waiting to test the result (steps 5-7). This invention proposes a way to let a user know much faster whether or not it is even possible that a question will have utility. Thus, if the question shows no value, there is no reason to add it to the pool of questions and iterate. The invention also proposes GUI elements that can help the user evaluate the utility of the question before having to iterate.

Advantages: The obvious advantage is that it will tremendously speed the development of applications using decision trees. There are 2 other advantages as well. First, it takes an experienced application developer to be able to use intuition to discover a good question to add. A novice user would waste lots of time adding bad questions and iterating. With this invention, many "bad questions" will not even be tried by the user. Second, this technique opens the door for an automatic tuning system (eg an expert system), which can automatically suggest questions, evaluate them, and test them. This would not be feasible if each question took 10 minutes to evaluate. But with this invention, only the most promising questions need to be examined, and these can be added in bulk (making use of the decision tree algorithms capability of "ignoring" useless questions). Thus with this invention, it is much more likely to make a self-tuning system.

2. How does the invention solve the problem or achieve an advantage,(a description of "the invention", including figures inline as appropriate)?

- Determine the potential usefulness of a new decision tree question without requiring the complete tree to be regrown:

This is done using the following techniques:

- Once the user proposes the new decision tree question, we can examine the existing decision tree to determine the path the incorrect answer took. Along this path, we know each question asked, and the conditional entropy drop attained by asking the OLD question at that node. At each node, we can evaluate the data at that node with the NEW question. If the conditional entropy drop by the NEW question is less, then we know that this question WOULD have been asked had it been available at the time the decision tree was built. It is possible the question could have been asked elsewhere, possibly hurting results in other sentences. But for this current error, the question does provide value. The real win comes in that often a question that seems to be valuable, provides no value for a specific problem. There are other questions which provide more information. Thus, the real win is when the question provides no value and does NOT provide a greater conditional entropy reduction than all the old questions. Also, generally the "higher" up in the decision tree a question is asked, the more important it is. Thus, the user gets feedback as to whether this is a really important question or one of lesser importance by how early in the tree this question would have been applied.
- Once one has confirmed that a question could provide value for this specific error, one can

examine its broader use by directly examining the complete set of sentences that illustrate the correct answer and the set of sentences that illustrate the incorrect answer. This is slightly different than the previous bullet in that some of these sentences might have been split off into a different part of the tree. The most significant a separation a question provides, the better. This helps one to select a better question if many pass the test provided by the first bullet.

Once the best question has been found, the complete tree can then be regrown, with greater confidence in knowing that it will be used to help solve the problem for which it is being added.

- Determine where in a decision tree the correct answer would be found, and show where the tree went awry for a particular error.

Sometimes when a decision tree system makes an error, it is not because of a missing question, but rather undertraining (which happens because of the data fragmentation done by the decision tree algorithm) or a bad question. This invention proposes a solution to this as follows:

- One can look at all leaves of the decision tree to search for the leaves that provide the "correct" answer. The correct answer is one in which the desired outcome has the greatest probability. These are the desired "target" leaves.
- One can then find the leaf that the incorrect answer reached.
- Then one can climb all these leaves upwards, looking for intersections where the leaves have common ancestors. The goal is to discover all nodes where the incorrect branch was taken. This finds that. By examining the entropy and probabilities of the correct child for each of these nodes, we can discover the more important nodes to focus on. The system can then let the user know which question was applied at that node, what the counts were, and even the sentences at the parent and 2 children.
- The GUI elements for viewing and debugging a decision tree.
Critical to a successful implementation of these algorithms is how the information is presented to the user. This invention proposes the following GUI look and feel:
 - A split screen display. One of the screens is the shape view, which shows the shape of the decision tree, along with the count of the number of events at each node.
 - In this view, the tree will be shown in one color, but the path taken from the root to incorrect leaf will be shown in red.
 - The correct leaves will be shown in green.
 - The common red ancestor nodes to green leaves will be shown with a different shape. Instantly, the user will be able to see a useful overview of where the correct parse needed to be, and where the incorrect parse ended.
 - The second window provides a more detailed zoom view of a single parent and its two children of the decision tree. This node in focus will be shown by drawing a box or circle around it in the shape view.
 - The detailed zoom window will show the entropy and histogram of the distributions for the parent and its children.
 - It can also show the smoothing lambdas.
 - By clicking appropriately, one can invoke a search for all sentences that contain data at that node.
 - By clicking appropriately, one can examine the question that is being applied at this node... its syntax, parameters, bitstrings, etc. One can also ask to examine where else in this or other decision trees this question is used.
 - One can also change the focus node by a specific click or keyboard shortcut. Thus one does not have to go back to the shape window to do this.

Eventually, this can be enhanced to provide a hint subwindow, which could search for questions that might be useful to drive the tree to learn the correct parse for this sentence.

- The ability to regrow only parts of a decision tree:

Once the user decides to add a question, it is not necessary to regrow the whole tree from scratch, though this is what is typically done. It is sufficient to adapt the current tree. One can apply this question at the root, and recursively down through its children. If ever this question provides a larger reduction in conditional entropy, then one can then apply this question at that node instead. Then, the complete subtree underneath this node will have to be regrown. But in general, a new question added to a model that is already functioning at 70-80%, will not be at the root or even 2nd level of the tree. Thus, if this question is only asked at one of the 4 grandchildren of the root, only 25% of the tree will have to be regrown. This can be a significant time savings. While researchers never worry about implementation details like this, the savings can add up when you're working to deploy a solution quickly for a customer on a tight schedule.

3. If the same advantage or problem has been identified by others (inside/outside IBM), how have those others solved it and does your solution differ and why is it better?

4. If the invention is implemented in a product or prototype, include technical details, purpose, disclosure details to others and the date of that implementation.

***Critical Questions (Questions 1 - 7 must be answered)**

*Question 1 On what date was the invention workable? 03/10/2000 Please format the date as MM/DD/YYYY (Workable means i.e. when you know that your design will solve the problem)

*Question 2 Is there any planned or actual publication or disclosure of your invention to anyone outside IBM?	<input type="radio"/> Yes <input checked="" type="radio"/> No
If yes, Enter the name of each publication or patent and the date published below. Publication/Patent: Date Published or Issued:	
Are you aware of any publications, products or patents that relate to this invention?	<input type="radio"/> Yes <input checked="" type="radio"/> No
If yes, Enter the name of each publication or patent and the date published below. Publication/Patent: Date Published or Issued:	

*Question 3 Has the subject matter of the invention or a product incorporating the invention been sold, used internally in manufacturing, announced for sale, or included in a proposal?	<input type="radio"/> Yes <input checked="" type="radio"/> No
Is a sale, use in manufacturing, product announcement, or proposal planned?	<input type="radio"/> Yes <input checked="" type="radio"/> No
If Yes, identify the product if known and indicate the date or planned date of sale, announcements, or proposal and to whom the sale, announcement or proposal has been or will be made. Product: Version/Release: Code Name: Date: To Whom:	
If more than one, use cut and paste and append as necessary in the field provided.	

*Question 4 Was the subject matter of your invention or a product incorporating your invention used in public, e.g., outside IBM or in the presence of non-IBMs?	<input type="radio"/> Yes <input checked="" type="radio"/> No
If yes, give a date. Please format the date as MM/DD/YYYY	

*Question 5 Have you ever discussed your invention with others not employed at IBM?	<input type="radio"/> Yes <input checked="" type="radio"/> No
If yes, identify individuals and date discussed. Fill in the text area with the following information, the names of the individuals, the employer, date discussed, under CDA, and CDA #.	

*Question 6 Was the invention, in any way, started or developed under a government contract or project?	<input type="radio"/> Yes <input checked="" type="radio"/> No <input type="radio"/> Not sure
If Yes, enter the contract number	

*Question 7 Was the invention made in the course of any alliance, joint development or other contract activities?	<input type="radio"/> Yes <input checked="" type="radio"/> No <input type="radio"/> Not Sure
If Yes, enter the following :Name of Alliance, Contractor or Joint Developer	
Contract ID number	
Relationship contact name	
Relationship contact E-mail	
Relationship contact phone	

Question 8 Have you submitted, or are you aware of, any related disclosure submission?	<input checked="" type="radio"/> Yes <input type="radio"/> No
If Yes, please provide the title and docket or disclosure number below:	
An Interactive Development Environment for Building High Quality Conversational Natural Language Applications	

Question 9 What type of companies do you expect to compete with inventions of this type? <i>Check all that apply.</i>
<input type="checkbox"/> Manufacturers of enterprise servers <input type="checkbox"/> Manufacturers of entry servers <input type="checkbox"/> Manufacturers of workstations <input type="checkbox"/> Manufacturers of PC's <input type="checkbox"/> Non-computer manufacturers <input type="checkbox"/> Developers of operating systems <input type="checkbox"/> Developers of networking software <input checked="" type="checkbox"/> Developers of application software <input checked="" type="checkbox"/> Integrated solution providers <input type="checkbox"/> Service providers <input type="checkbox"/> Other (Please specify below)

Patent Value Tool (Optional - this may be used by the inventor and attorney to assist with the evaluation)

(The Patent Value tool can be used by you or the evaluation team to determine the potential licensing value of your invention.)

These are the answers which were entered into the **Patent Value Tool**.

Market

What is the anticipated annual market size (in dollars) that will be captured by your invention?

\$100M to \$1B

Reason(s) for above Answer Decision Trees are used in many applications. This speeds development of decision tree models.

CLAIMS

Question 1 - How new is the technical field?

Emerging

Reason(s) for above Answer While decision trees have been around for more than 20 years, they are just starting to get deployed in NLU systems.

Question 2 - How central is the invention to the product(s) which might be expected to contain the invention?

Main

Reason(s) for above Answer This is not necessary to use decision trees... it will just speed development.

Question 3 - What is the scope of the claim?

Fundamental

Reason(s) for above Answer This idea will work in any decision tree application.

PORTFOLIO NEED

PPM Needs

What are the portfolio needs in the area of your invention?

Listed in PPM Needs

EXPLOITATION & ENFORCEMENT

Question 1 - How easily can the use of the invention by a competitor be detected?

Trivially

Reason(s) for above Answer Since this invention proposes not only algorithms, but a look and feel interface, this can easily be spotted.

Question 2 - How easily can the use of the invention be avoided by a competitor?

With much work

Reason(s) for above Answer Other techniques could be used, like completely regrowing the tree each time a question is added. It just puts the burden on the user. Thus, while it can easily be avoided, doing so significantly impedes the competitors ability to develop a timely solution for a customer.

BUSINESS VALUE

Question 1 - What percentage of the companies producing products in the field of this invention might use this invention?

By 10% to 30%

Question 2 - What is the value of this patent to current or anticipated Alliance Activity between IBM and other companies?

High value

Question 3 - What is the value of this patent to current or anticipated Technology Transfer Activity between IBM and other companies?

High value

Question 4 - Does it result in prestige to IBM?

Industry wide

Reason(s) for above Answer This will help people in Computational Linguistics realize that decision trees are easy to use and tune. Even those experienced will appreciate the contributions.

Post Disclosure Text & Drawings

(Form Revised 12/17/97)